RECEIVED
CENTRAL FAX CENTER
NOV 08 2005

Doc Code: AP.PRE.REQ

PTO/SB/33 (07-05)
Approved for use through xx/xx/200x. OMB 0651-00xx
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional): 200308328-1

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]

on 11-8-05

Signature: Carrie McKerley

Typed or printed name: Carrie McKerley

Application Number	Filed
10/005,923	Dec 4, 2001

First Named Inventor: Cormac A. Flanagan

Art Unit	Examiner
2192	John J. Romano

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
Note: No more than five (5) pages may be provided.

I am the

- [ ] applicant/Inventor.
- [ ] assignee of record of the entire interest. See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)
- [X] attorney or agent of record. Registration number 40,709
- [ ] attorney or agent acting under 37 CFR 1.34. Registration number if acting under 37 CFR 1.34 _____

Signature

Philip S. Lyren
Typed or printed name

281 514 8236
Telephone number

11-8-05
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below*.

[X] *Total of 1 forms are submitted.

RECEIVED
CENTRAL FAX CENTER

NOV 08 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicants:	Conmac A. Flanagan et al.	Examiner: John J. Romano
Serial No.:	10/005,923	Group Art Unit: 2192
Filed:	December 4, 2001	Docket No.: 200308328-1
Title:	Method and Apparatus for Automatically Inferring Annotations	

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants request review of the Final Office Action of August 10, 2005 in the above-identified patent application. Applicants present three separate issues (I – III) below for review.

ISSUE I: Claim Rejections: 35 USC § 103

Claims 1-8, 14-28, 32-42, and 46-51 are rejected under 35 USC § 103(a) as being unpatentable over USPN 5,423,027 (hereafter Jackson) in view of USPN 6,343,376 (hereafter Saxe). Applicants respectfully traverse.

The present application has five independent claims (1, 22, 36, 50, and 51). For discussion, Applicants argue claim 1 to show how the limitations of this claim are not taught or suggested in Jackson in view of Saxe. The allowance of the other independent claims (22, 36, 50, 51) will be apparent.

Overview of Claim 1

Claim 1 teaches a program checking tool that discovers warnings in a computer program. Once a warning is discovered, it is mapped to an annotation in the computer program. The program checking tool then modifies the computer program so the warning is not produced. These steps of finding warnings, mapping warnings to annotations, and modifying the program are repeated until no warnings are produced. A user is then provided with the modified computer program having no warnings.

Overview of Jackson

By way of background, computer programs are often embellished with specifications that specify dependencies among different components of objects in the program (col. 2, lines 3-37). "These specifications typically take the form of lines added into the code by a programmer or by the checker program" (col. 3, lines 38-40).

Jackson teaches a checker program that finds errors in computer programs (see col. 4, lines 12-18). Two important questions are: How does Jackson's checker program find errors, and what does the checker program do with the errors once they are discovered. Jackson expressly answers these questions.

Jackson generally teaches three steps for locating and reporting errors. First, the programmer or the checker program adds aspect specifications to the computer program (col. 5, lines 63-67). Second, checker program reviews the aspect specifications and program code to locate errors (col. 5, line 67 – col. 6, line 2). Third, the program checker outputs the identified errors to a user (col. 6, lines 2 – 3).

Argument A

Applicants argue that the teachings in Jackson are very different than the recitations in claim 1. Applicants reiterate that Jackson teaches three steps for locating and reporting errors: (1) use program checker to add lines of aspect specification to a computer program, (2) use program checker to process the computer program and find errors, and (3) report errors to a user. For numerous reasons, these steps are quite different than the steps in claim 1. First, claim 1 recites (a) applying a program checking tool to the computer program to produce a warning. Notice that step (a) in claim 1 is the third step in Jackson. Then, claim 1 recites (b) mapping the warning to an annotation modification. Jackson never performs this step! In Jackson, once an error is discovered, it is provided to a user. Then, claim 1 recites (c/d) modifying the computer program until no warnings are produced. Jackson never performs this step!

Once the checker program in Jackson discovers an error, the error is reported to a user. Jackson does not teach that the checker program maps a discovered error to an annotation and then modifies the annotation to cure the error. FIG. 11 in Jackson is a flow diagram that teaches what the checker program does with errors once they are discovered. Applicants reproduce portions of Jackson's specification that discuss FIG. 11:

> **When the aspect view specification and the flow specification are completed, a programmer passes the entire annotated program or only a portion of the program through the checker to check for errors.** The checker generates both error messages and warning messages. The error messages indicate faults in the aspect specification text. These may be lexical errors, syntax errors or violations of consistency checks. Warning messages indicate faults in the aspect specification text such as the failure to include all the aspects of results objects and in flow specification. Conjecture messages, on the other hand, note discrepancies between the code and the aspect specification. The conjecture messages serve to identify the missing dependency and to indicate procedure wherein the error is found.
>
> **The checker locates errors by the method described in FIG. 11** by initially constructing a flow for the code in step 160 and comparing the constructed flow with the specified flow is step 162. **The checker reports an error in step 164 if the constructed flow does not correspond to the flow specification.** (Emphasis added: Col. 13, lines 21 – 42).

Thus, Jackson expressly teaches that the checker program locates errors in the computer program and then reports these errors to the user. Nowhere does Jackson teach or suggest that the checker program maps the errors to annotations then modifies the annotations to cure the errors. Specifically, the flow diagram in FIG. 11 of Jackson expressly shows that once errors are reported in block 164, flow ends (i.e., the arrow from block 164 links to the "end" block).

For at least these reasons, independent claims 1, 22, 36, 50, and 51 are allowable over Jackson in view of Saxe. The dependent claims are allowable for at least the reasons given in connection with the independent claims.

Response to Examiner's Argument

The Examiner argues that Jackson's step of using the program checker to add lines of aspect specification to a computer program is equivalent to the claim recitation of "mapping a

warning into an annotation modification" (see FOA at pages 3-4). Applicants respectfully disagree because the express language of Jackson teaches that errors are sent to a user, not mapped to annotations. The Examiner's interpretation is contrary to the language in Jackson itself (see FIG. 11 of Jackson, the summary, the specification at col. 13, lines 21 – 42, etc.).

Further, the Examiner repeatedly cites Jackson at col. 5, lines 63-67. Applicants respectfully encourage the Board to read this citation. This section of Jackson actually supports the position of the Applicants. Namely, Jackson teaches three steps for locating and reporting errors: (1) use program checker to add lines of aspect specification to a computer program, (2) use program checker to process the computer program and find errors, and (3) report errors to a user. Nowhere does this section of Jackson teach or suggest mapping errors to an annotation and then modifying the annotation/program to cure the errors. In Jackson, errors are sent to a user.

Argument B

The Office Action admits that "Jackson does not expressly disclose repeating each of steps a, b and c until no warnings produced" (see OA p. 4). Applicants agree with this admission. The Office Action, however, attempts to cure this deficiency with Saxe. Applicants respectfully disagree.

First, Jackson actually teaches away from repeating each of steps a, b, and c until no warnings are produced. The specification in Jackson (example, see FIG. 11 and col. 13, lines 21 – 42) teaches that errors are sent to the user, not cured with the checker program.

Second, the Office Action cites Saxe at col. 7, lines 19-26 for teaching the limitation of d) repeating each of steps a, b, and c. This section of Saxe is reproduced below for convenience:

> In an extension of the present invention, detection by the prover 204 of an error or inconsistency in the formula being analyzed permits the output of the prover to be used as the basis for assisted amelioration or elimination of the diagnosed formal error, as shown at 209. The edited source code may optionally be fed back to the verification condition generator 202, causing the process to repeat from the beginning as described earlier.

This section of Saxe merely teaches that a section of source code is fed back into a verification condition generator. By contrast, claim 1 recites repeating each of three steps: (a) applying a checking tool to produce one or more warnings, (b) mapping at least one warning into an annotation modification, and (c) modifying the computer program in accordance with the annotation modification. Nowhere does Saxe teach or suggest that steps a, b, and c of claim 1 are repeated until no warnings are produced. Instead, Saxe merely teaches that a section of source code is fed back into a verification generator.

ISSUE II: Claim Rejections: 35 USC § 103

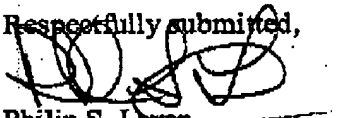
Claim 9 is rejected under 35 USC § 103(a) as being unpatentable over Jackson in view of Saxe and further in view of USPN 6,553,362 (hereafter Saxe 362). Applicants respectfully traverse. Dependent claim 9 indirectly depends from independent claim 1. Thus, for at least the reasons given above in connection with independent claim 1, claim 9 is allowable.

ISSUE III: Claim Rejections: 35 USC § 103

Claims 10-13, 29-31, and 43-45 are rejected under 35 USC § 103(a) as being unpatentable over Jackson in view of Saxe and further in view of USPN 6,154,876 (hereafter Haley). Applicants respectfully traverse. Dependent claims 10-13, 29-31, and 43-45 depend from independent claims 1, 22, and 36. Thus, for at least the reasons given above in connection with the independent claims, these dependent claims are allowable.

For at least these reasons, Applicants request withdrawal of the final rejection.

Respectfully submitted,



Philip S. Lyren
Reg. No. 40,709
Ph: 281-514-8236

CERTIFICATE UNDER 37 C.F.R. 1.8

The undersigned hereby certifies that this paper or papers, as described herein, is being transmitted to the United States Patent and Trademark Office facsimile number 571-273-8300 on this 8th day of November, 2005.

By Carrie McKerley
Name: Carrie McKerley